

A semi-supervised learning approach towards automatic wireless technology recognition

Miguel Camelo*, Adnan Shahid†, Jaron Fontaine†, Felipe Augusto Pereira de Figueiredo†, Eli De Poorter†, Ingrid Moerman†, Steven Latre*,

* imec - IDLab, Department of Mathematics and Computer Science, University of Antwerp

† imec - IDLab, Department of Information Technology at Ghent University

Email: miguel.camelo@uantwerpen.be

Abstract—Radio spectrum has become a scarce commodity due to the advent of several non-collaborative radio technologies that share the same spectrum. Recognizing a radio technology that accesses the spectrum is fundamental to define spectrum management policies to mitigate interference. State-of-the-art approaches for technology recognition using machine learning are based on supervised learning, which requires an extensive labeled data set to perform well. However, if the technologies and their environment are entirely unknown, the labeling task becomes time-consuming and challenging. In this work, we present a Semi-supervised Learning (SSL) approach for technology recognition that exploits the capabilities of modern Software Defined Radios (SDRs) to build large unlabeled data sets of IQ samples but requires only a few of them to be labeled to start the learning process. The proposed approach is implemented using a Deep Autoencoder, and the comparison is carried out against a Supervised Learning (SL) approach using Deep Neural Network (DNN). Using the DARPA Colosseum test bed, we created an IQ sample data set of 16 unknown radio technologies and obtain a classification accuracy of $> 97\%$ using the entire labeled data set using both approaches. However, the proposed SSL approach achieves a classification accuracy of $\geq 70\%$ while using only 10% of the labeled data. This performance is equivalent to 4.6x times better classification accuracy than the DNN using the same reduced labeled data set. More importantly, the proposed approach is more robust than the DNN under corrupted input, e.g., noisy signals, which gives us to 2x and 3x better accuracy at Signal-to-Noise Ratio (SNR) of -5 dB and 0 dB, respectively.

Index Terms—wireless technology recognition, semi-supervised learning, deep learning, neural network, deep autoencoders.

I. INTRODUCTION

According to Cisco, global mobile data traffic will increase seven-fold between 2017 and 2022 and thereby, increasing the Compound Annual Growth Rate (CAGR) of 46 percent [1]. Such increased traffic cannot be accommodated even by spectrum extension. In this regard, the radio spectrum becomes a valuable entity since many wireless technologies share the same spectrum. However, most of the spectrum is underutilized most of the time, while radio technologies suffer poor performance due to interference. In such a coexisting environment, Cognitive Radio (CR) systems will play a significant role to solve this problem [2]. Therefore, providing intelligence to the radios, so that they can reason about how to use and share the available spectrum efficiently, and defining new spectrum access strategies is of fundamental importance.

Inside CR, Dynamic Spectrum Access (DSA) provides the capability to share the spectrum among multiple technologies

in an opportunistic manner. One critical problem that DSA faces is to identify if some technology is accessing the same spectrum and then take appropriate measures to combat the performance degradation due to interference. This problem is termed as the Technology Recognition (TR) problem, and it refers to identify radio signals of wireless technologies without requiring any signal pre-processing such as channel estimation, and timing and frequency synchronization [3].

Traditionally, TR is done by domain experts, which use carefully designed hand-crafted rules to extract features from the radio signals. On the contrary, state-of-the-art approaches based on DNNs can extract features directly from raw input data and perform the recognition task on those features automatically. However, DNNs approaches have two main drawbacks: 1) they are mainly trained in a supervised way, which implies that the whole data used for training must be labeled, 2) their training algorithms, such as Stochastic Gradient Descent (SGD) [4], require a large amount of data to obtain a good performance [5], otherwise the resulting trained model may suffer severe overfitting problems [6].

Generally, assigning labels to data can be expensive, e.g., very time-consuming, and/or some of the data might not have any labels at all due to incomplete knowledge of the ground truth class labels, e.g., the radio technologies to be classified are entirely unknown. On the contrary, sensing the spectrum using modern radios allows collecting a large amount of unlabeled data available at no cost. Therefore, formulating the TR problem that uses both labeled and unlabeled data and designing robust systems that can deal with different amounts of them is of utmost importance.

SSL is a Machine Learning (ML) technique that learns from unlabeled data by extracting a good representation of the data distribution and then use it to solve the supervised problem with a reduced number of labels [7]. Given the number of wireless technologies that already exist and the new ones which are under development, there is a need for efficient usage of the spectrum via collaboration and coexistence. For this, CR systems require a new TR approach that provides the flexibility of exploiting a large amount of unlabeled data while improving the classification accuracy by using a limited labeled data set.

In this paper, we propose a SSL-based TR approach that addresses these new requirements of CR systems. The main

contributions of our work are threefold. First, we introduce a SSL approach for wireless technologies recognition that can work on raw In-phase and Quadrature (IQ) samples and does not require the whole data set to be labeled, which is a time-consuming and a challenging task. Second, the proposed scheme is implemented using Deep Autoencoder (DAE), which requires an unlabeled data set and only a few labeled examples. The performance of the scheme is evaluated against a DNN architecture that requires the whole data set to be labeled. We show that the proposed scheme outperforms the DNN, while still requires limited labeled data set. Finally, the SSL approach was evaluated in the DARPA Colosseum testbed [8], and it was able to successfully recognize sixteen different unknown wireless technologies with an accuracy of above 97% using the entire data set, and $> 70\%$ using only 10% of it. This translates to 4.6x times better accuracy than the DNN model using the same amount of labeled data. Using the DARPA Colosseum, we provide strong evidence about the robustness of the proposed approach.

The rest of this paper is structured as follows. Recent works on radio signal identification are discussed in Section II. The TR problem formulation is described in Section III. Section IV presents the proposed approach to solve the TR problem, and Section V contains the details about the data set used during evaluation. Performance evaluation results are provided in Section VI, and Section VII concludes the paper.

II. RELATED WORK

In this section, we present some of the most relevant work on TR. For a more exhaustive review on the general radio signal identification problem, we recommend [9] and [10] to the readers.

TR has been mainly applied to the identification of communication systems based on the differentiation of their channel method access, e.g., Single Carrier (SC) vs. Multiple Carrier (MC) [11], and it has been extended to classify various wireless communication technology standards, e.g., WiMAX vs. LTE [12]. Similar to other related tasks in radio signal identification, traditional approaches for TR are based on Likelihood-Based (LB), and Feature-Based (FB) using high-order statistics features such as moments, cumulants and cyclic cumulants. Karami et al. propose an algorithm to identify Spatial Multiplexing (SM) and Alamouti (AL)-coded Orthogonal Frequency Division Multiplexing (OFDM) signals for Multiple-Input and Multiple-Output (MIMO) systems based on second-order signal cyclostationarity [13]. This algorithm only requires the cross-correlation of the received signals on multiple antennas to discriminate between these two classes of OFDM systems. Firdaoussi et al. propose a method to obtain the Generalized Mean Ambiguity Function (GMFA) of the received signal and use it to discriminate between OFDM signals and Single Carrier Linear Digital (SCLD) in channels with additive white Gaussian noise [14].

For identification of wireless technology standards, Bouzegzi et al. propose an algorithm that can discriminate among different technologies such as WiMax, WiFi, and

DVBT by exploiting the fact that these technologies are based on OFDM but differ from their intercarrier spacing used in OFDM [15]. The algorithm estimates the intercarrier spacing based on the maximum-likelihood principle. The proposed algorithm does not need a training sequence and is more robust than autocorrelation-based methods under small length cyclic prefixes and multipath environment. Al-Habashna et al. propose an algorithm based on second-order cyclostationarity properties of the LTE and WiMAX technologies and use them as discriminating features for classification [12]. The proposed algorithm does not require a carrier, waveform, and symbol timing recovery information. This approach also provides immunity to phase, frequency, and timing offsets. The referred approaches for TR require expert knowledge for either modeling the signals and the environment (LB methods) or selecting the required features (FB methods). Therefore, they cannot be used for identification of unknown radio technologies.

More recently, several approaches based on Deep Learning (DL) has been proposed to solve the TR task using raw time, frequency, and time-domain data. Kulin et al. propose a Convolutional Neural Network (CNN) to identify single transmissions of ZigBee, WiFi, and Bluetooth radio technologies using raw IQ and the Fast Fourier Transform (FFT), the amplitude and phase of the raw IQ samples [16]. Without requiring any feature engineering, the proposed models achieved an accuracy above 80% in scenarios with $\text{SNR} > -10\text{dB}$, and above 95% accuracy using raw IQ samples, and near 100% with FFT and amplitude/phase in scenarios with $\text{SNR} > 5\text{ dB}$.

Biter et al. propose a CNN that can recognize 802.x standard compliant technologies using time-frequency representation of the spectrum for a wide range of SNRs [17]. This model outperforms standard feature based classification methods in terms of classification accuracy, and it is also able to detect and identify these technologies when they are overlapping in time. Finally, Yi et al. propose a real-time external interference source classification method for a ZigBee-based wireless sensor network using a CNN as a classifier and Received Signal Strength Indication (RSSI) values as input data [18]. As interference, the model can identify WiFi beacons, WiFi video streaming, WiFi file transfer, Bluetooth iBeacon, and a microwave oven RSSI traces. The proposed model can achieve an accuracy of over 93% detecting the different classes of interference with minimal computational resources.

In general, we can see that traditional methods such as LB and expert FB engineering combined with pattern recognition have been outperformed by supervised DL methods in the task of TR. Supervised DL methods remove the need for expert knowledge about the environment and the signal features used for classification by using the power of automatic feature abstraction. However, it requires the whole data set to be labeled. In the case of both the technologies to be recognized and the environment are entirely unknown, the labeling task becomes time-consuming and challenging. To overcome these limitations, in this paper, we propose a SSL approach for TR that separates the feature extraction from the classification

task in the DL architecture, so the use of unlabeled data is maximized. At the same time, the proposed approach minimizes the use of domain expertise knowledge by requiring only a small portion of the entire data set to be labeled to obtain a good performance, which is not the case of supervised DL models.

III. SPECTRUM MANAGER FRAMEWORK AND PROBLEM FORMULATION

In this section, the proposed spectrum manager framework and mathematical formulation of the Technology Recognition (TR) problem are presented.

A. Spectrum Manager Framework

Figure 1 shows a spectrum manager framework which elaborates where the results of TR can be used. The framework comprises of a spectrum manager, which makes spectrum decisions, and N radios, which represent unknown wireless technologies. The goal of the spectrum manager is to assist the N unknown wireless technologies to make spectrum decisions by first identifying them and then doing frequency domain analysis. In order to enable this, the spectrum manager executes the following tasks in the listed manner: a) *training*, b) *validation*, c) *frequency domain analysis*, and d) *spectrum decision*. In this work, we focus on the lower two blocks, i.e., *training* and *validation*, to enable TR for CR systems.

Focusing on the TR block, the *training* task is used to train a model in a semi-supervised way with raw IQ samples of the N radios using a DAE. The detailed description of the semi-supervised approach and its implementation is given in Section IV and Section V, respectively. Once the model is trained, in the *validation* task, it can identify the N unknown wireless technologies. In the *frequency domain analysis* task, frequency domain analysis of the identified technologies is done by extracting spectrum occupancy information of the technologies. Finally, in the *spectrum decision* task, the radio uses the extracted spectrum efficiency information to define actions, such as change the center frequencies of the radios and assign a collision-free time slot for transmissions, so that fair coexistence can be realized. Once the spectrum decisions are made, they are notified to the N radios via control channels.

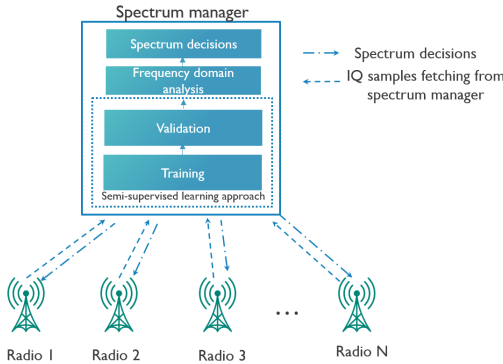


Fig. 1: Proposed framework.

B. Problem Formulation

We consider a communication system in which the received signal $r(t)$ can be represented as follows:

$$r(t) = s(t) * h(t) + \omega(t), \quad (1)$$

where $s(t)$ is the original transmitted signal, $h(t)$ is the time varying impulse response of the transmit channel, and $\omega(t)$ represents Additive White Gaussian Noise (AWGN) with zero mean and variance σ^2 . In modern digital communication systems, the transmitted signal $s(t)$ is modeled as follows:

$$\begin{aligned} s(t) &= I(t)\cos(2\pi f_c t) + Q(t)\sin(2\pi f_c t), \\ s(t) &= i(t) + jq(t) \end{aligned} \quad (2)$$

where $s(t)$ is called Quadrature signal or IQ samples, and the $i(t)$ and $q(t)$ are termed as the *in-phase* and *quadrature* components, respectively. In practice, this approach is widely used due to its simplicity for representing mathematical operations and its flexibility to generate any modulation scheme based on different values of $I(t)$ and $Q(t)$.

Given a classification problem with an input vector set X and their corresponding target variables set Y , the objective is to find a function f that predicts $y \in Y$ given a new value for $x \in X$, where y represents L class labels.

$$\begin{aligned} f : \mathbb{R}^n &\rightarrow 1, \dots, K \\ y &= f(x) \end{aligned} \quad (3)$$

When the function f is used to map a given signal $s(t)$ to a set S of signal classes (labels) without requiring any pre-processing of the signal, then the classification problem is termed as Automatic Signal Identification (ASI) [3]. Traditionally, this problem has been studied in two main research lines: Automatic Modulation Classification (AMC), where Y is the set of modulation schemes to be identified, and TR, where Y is the set of wireless radio technologies to be identified, e.g., generic medium access technologies such as SC vs MC, standard wireless technologies such as GSM, WiMAX, LTE, WiFi, Bluetooth, and ZigBee, among other, or more recently, unknown technologies such as the ones participating in the DARPA Spectrum Collaboration Challenge (SC2) competition.

In ASI, three approaches have been mainly used in the literature: Likelihood-Based (LB), Feature-Based (FB), and supervised Deep Learning (DL). The first two use signal processing and pattern recognition methods for the identification of signals. On the other hand, DL methods are Artificial Intelligence (AI) algorithms that learn by representing the world as a nested hierarchical of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones [6]. DL algorithms allow building the mathematical function f as a combination of many simpler functions. Up to date, most of the research on TR has been focused on using Deep Neural Networks (DNNs) that are trained in a supervised way. However, this method has a drawback: building f for a complex task as TR requires a large data set of training examples with their

corresponding labels. Otherwise, the training with supervised learning techniques on a small labeled data set often results in learning the training data but severely failing on predicting the correct class of unseen data (overfitting) [6].

In real environments, we often have the ability to collect large amounts of (unlabeled) IQ samples, but labeling all of them may be costly, e.g., in terms of time, and/or some of the data might not have any labels at all due to incomplete knowledge of the ground truth class labels, e.g., the radio technologies to be classified are entirely unknown and increases the complexity of the labeling task. Therefore, solving the TR problem for unknown technologies in unknown environments, which is the case of SC2 competition, is challenging and requires another approach that allows DL models to use unlabeled data to bootstrap the learning and minimizing the number of labels required to solve the supervised learning task efficiently.

IV. A SEMI-SUPERVISED SYSTEM FOR TR

In this section, we present the proposed SSL approach by first introducing the core system along with the description of the main components.

A. Learning with a few labels

Let $X = \{x_1, x_2, \dots, x_N\}$ and $Y = \{y_1, y_2, \dots, y_N\}$ be the sets of N examples and their corresponding labels, respectively, where $x_i \in X$ and $y_i \in Y$ for all $i \in [N] := \{1, 2, \dots, N\}$. Unsupervised Learning (USL) algorithms have the goal of finding interesting features from the unlabeled data X . In SL, the goal is to learn a mapping from X to Y given a training set of pairs (x_i, y_i) , where y_i is the label of the i th example x_i . Semi-supervised Learning (SSL) is an approach that falls between USL and SL. In this family of learning algorithms, the set X is divided in two subsets $X_s = \{x_1, x_2, \dots, x_L\}$, for which their corresponding labels $Y_s = \{y_1, y_2, \dots, y_L\}$ are provided, and $X_u = \{x_{L+1}, \dots, x_N\}$, for which no labels are provided such that $X = \{x_1, x_2, \dots, x_L, x_{L+1}, \dots, x_N\}$.

To use SSL algorithms for recognition, it is required that the knowledge acquired about the distribution of the examples from the unlabeled data set, i.e., $p(x)$, is useful to infer $p(y|x)$. Otherwise, semi-supervised learning may decrease the performance of the SL classifier by misguiding it during the learning process. SSL tries to use the unlabeled data to learn valuable information about the data, and then use it to fine-tune a classifier with a reduced number of labels. By extending this approach and generalizing it to be used as a system for TR, we can use a reduced number of labels, in comparison to a DL model, with good accuracy on the recognition task, even if the technologies are entirely unknown and no information about the environment is provided.

B. System description

Sensing and capturing over-the-fly radio signals in the form of IQ samples is a simple task and can be performed using SDR platforms. However, IQ samples labeling is a difficult task due to the following reasons: 1) an expert is

required to identify and label each captured sample and 2) in unknown environments, the number of unknown signals increases the complexity of the labeling task. The proposed approach decouples the feature extraction via unsupervised learning, and the classification tasks via supervised learning while keeping the high expressiveness of DL models. The overall workflow of the proposed semi-supervised learning approach is shown in Figure 2. A description of each block and the details of the actual implementation are given below.

Spectrum sensing: This module is in charge of sensing the spectrum and capturing IQ samples that will be further processed by the subsequent blocks.

Data Transformation: Depending on the model to be trained, the original IQ samples, which are time domain representation of radio signals, can be transformed into other domains such as frequency or time-frequency. In this work, we focus on the IQ samples representation as it does not require any further processing.

Data labeling system: In this block, two steps are performed: a) samples selection and b) labeling of the samples. Since the proposed architecture is semi-supervised, it is essential to select representative samples of the radio technologies that need to be identified. Here, domain expert knowledge or in combination with pseudo labeling is used. This block stores all the samples and the labels associated with the labeled samples.

Data storage: This block is composed of two databases: 1) sample database and 2) label database. IQ samples are stored in the sample database, while the label database is used for storing the labels of a reduced set of examples. Depending on the kind of data and the training strategy, the databases are connected to one or more blocks: the supervised learning (sample database and label database), the unsupervised learning (sample database), and the batch system (sample database and label database).

Batch system for online training: In the offline training, the input data is created by selecting a portion of the data from the sample database via some predefined strategy, e.g., uniform random selection. In the online training, on the other hand, the input can be provided by a batch system that takes data from the sample database and uses it for retraining a model.

Semi-supervised classification: This block receives the sensed data and performs the classification task. The block also receives a limited labeled data set from the data labeling system block. Based on the labeled and unlabeled data sets, different learning algorithms can be used in the supervised and unsupervised learning blocks, and how they interact to perform the SSL task.

Technology Recognized: This is where the proposed architecture indicates to which class a given capture sample belongs to. Note that a class label may be as simple as the name of the technology, but it can also be more expressive and contain information about spectrum utilized over time, central frequencies, duty cycle, etc.

The proposed workflow is flexible to support a range of SSL algorithms, training methods, and input types. The selection

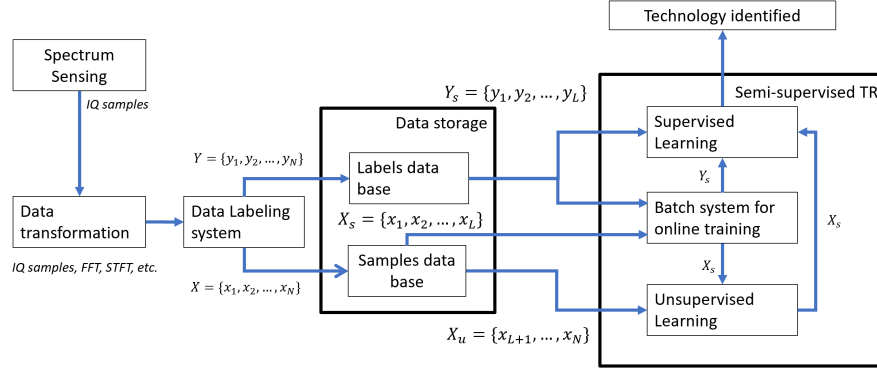


Fig. 2: Workflow of the proposed semi-supervised learning approach for TR.

of the semi-supervised approach mainly depends on various factors including the amount of available data, the number of labels, the complexity of the radio signals to be identified, and the need for offline or online training capabilities, etc.

C. SSL using Deep Autoencoders (DAEs)

The SSL TR block showed in Figure 2 was implemented using a DAE [19]. The resulting architecture of the DAE for TR is shown in Figure 3. DAEs are DNNs that are trained to copy its input to its output. A DAE is composed of two parts, an encoder that maps $h = f(x)$, where h is known as the code, and a decoder that produces a reconstruction $r = g(h)$. In practice, DAEs are not trained to get $x = g(f(x))$ but to obtain an h that contains only useful information about x . To do that, h is constrained in such a way that its dimension is smaller than x . This kind of DAE is called under-complete. The learning process of a DAE can be defined as:

$$\text{minimize } L(x, g(f(x))) \quad (4)$$

where L is a loss function indicating how similar is the input x and the reconstructed output $g(f(x))$. The idea of under-complete DAE is to learn only important features of the data distribution. To enforce learning good features and avoid learning to copy the input to the output, denoising DAE uses a different loss function to discourage learning the identity function as follows:

$$\text{minimize } L(x, g(f(\bar{x}))) \quad (5)$$

where \bar{x} is a copy of x that has been corrupted with some noise. In this way, the DAE does not learn to map $x \rightarrow x$ but undo the corruption by learning the structure of $p(x)$ [20].

For SSL, DAE provides a two-step training process: First, we train the DAE M_{AE} , which is composed of the encoder M_E and decoder M_D networks, in an unsupervised way using only X_u . Second, after the unsupervised learning, we create a train a classifier M_{EC} using the encoder M_E together with a SoftMax classifier M_C in a supervised way using the reduced labeled data set X_s .

During the supervised training, M_E is used as a feature extractor for M_C . This step provides an initial bootstrapping on the classification task. Then, a fine-tuned step is performed, i.e., all layers in M_{EC} are retrained, in order to increase the

accuracy of the resulting model. Algorithm 1 shows the pseudo code of the training procedure.

Algorithm 1: SSL procedure using DAE

Input:

- Unlabeled dataset: X_u
- Labeled dataset: X_s
- (Optional) Trained Autoencoder network: M_{AE}
- (Optional) Trained Encoder-Classifier network: M_{EC}

Output:

- Trained models: M_{AE} and M_{EC}

if M_{AE} *Exists* **then**

 UnFreezeEncoderWeights(M_{AE});

else

$M_{AE} = \text{CreateModel}(M_E, M_D)$;

while *Unsupervised training* **do**

 TrainAutoencoder(M_{AE}, X_u);

if M_{EC} *do not Exist* **then**

$M_{EC} = \text{CreateModel}(M_E, M_C)$;

while *Supervised training* **do**

 FreezeWeightsEncoder(M_{EC});

 TrainClassifier(M_{EC}, X_s);

 UnFreezeWeightsEncoder(M_{EC});

 TrainClassifier(M_{EC}, X_s);

return M_{AE} and M_{EC}

The DAE block of our system was designed by following a CNN architecture [21]. While traditional DNNs are built by connecting a series of fully-connected layers, CNN connects the neurons of a given layer, called Conv layer, with only a few numbers of neurons of the next layer to reduce the computational complexity of the learning. Note that this kind of networks has been shown to perform well with IQ samples as input [22], which motivates us to follow the same design pattern.

The encoder of the DAE is composed of two Conv layers with rectified linear unit (ReLU) activation function, each one followed by a batch normalization and a dropout layer for regularization. We use strides > 1 instead of max-pooling layers. Note that the dropout layers allow the DAE to behave as a denoising DAE to improve its capacity as feature extractor [23]. Figure 3 shows an overview of the resulting architecture and the parameters of the Conv layers. The specific parameters of each layer, such as the number of filters, strides, dropout rate, etc., were determined using a hyperparameter swapping. The proposed encoder configuration generates an intermediate

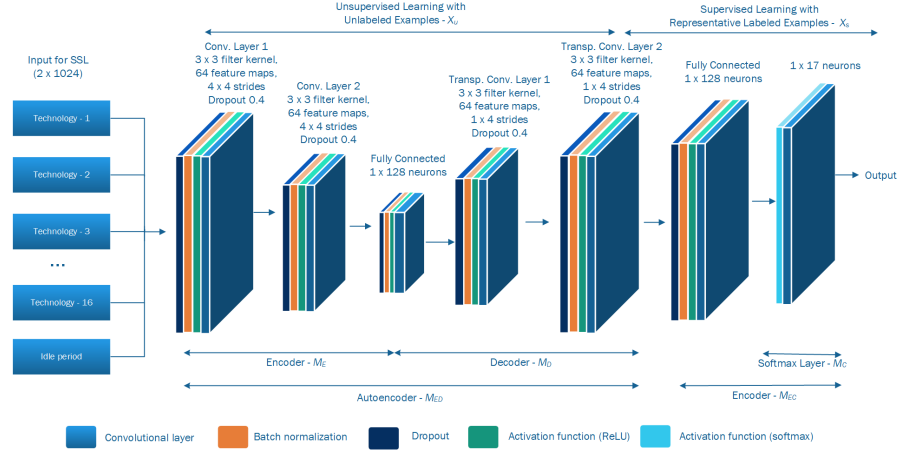


Fig. 3: SSL algorithm implemented using DAE

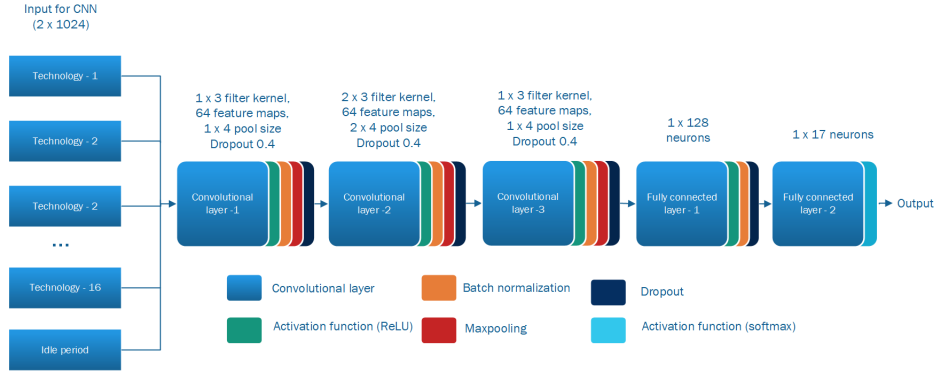


Fig. 4: Base-line SL algorithm implemented using CNN

code of size 128, e.g., a reduction factor of 16x.

Similarly, the decoder part follows the same pattern but in reverse order and replacing the Conv layers by Transposed Conv. The DAE contains 1M of trainable parameters. The Autoencoder (AE) was trained by using batches of size 128, the Adam optimizer [24] with a learning rate of 0.0004, and binary cross-entropy as the loss function for reconstruction. We implement our model in Keras [25] with TensorFlow [26] as the back-end, and it was training during 200 epochs: 100 epochs in unsupervised mode, 50 epochs only classifier (phase 1), and 50 epochs fine-tuning (phase 2).

The supervised part of the architecture is composed of the encoder part of the DAE in addition to two dense layers, one with 128 neurons, and the second one with 17 neurons and a SoftMax activation layer for classification. The resulting model (encoder+dense layers and classification) has 500k and 18k trainable parameters in phase 1 and phase 2, respectively. In this model, This model was trained using the same parameters as the DAE except that the loss function was categorical cross-entropy, and the learning rate was reduced to 0.004.

D. Baseline using CNN

We implemented and trained a CNN architecture to be used as a baseline for comparison with the proposed SSL approach using DAE. This architecture follows is composed of three Conv layers with rectified linear unit (ReLU) acti-

vation function, each one followed by a max-pooling, batch normalization, and a Dropout layer for regularization. The CNN model was fine-tuned to have a high performance using the entire data set, and it was trained during 100 epochs to guarantee a fair evaluation. Figure 4 shows an overview of the resulting CNN model.

V. DATASET GENERATION

We generated the data set in the DARPA Colosseum [8], the testbed used for the DARPA's three-year SC2s competition on smart radios and spectrum sharing. The SDRs available in Colosseum are Ettus Universal Software Radio Peripheral (USRP) X310 with UBX 160 USRP Daughterboards [27]. Given this hardware configuration, the Radio Frequency (RF) monitor module is implemented as a thread running along with the Physical Layer (PHY) of our radio stack. The RF monitor uses the USRP's Radio stream # 1, where the RX channel is used for spectrum sensing, and the TX channel is used to support dual PHY transmissions. Figure 5 shows how the RF monitor block interacts with the USRP and TR blocks. Note that this module exclusively serves the ML algorithms running in our radio, and it sends the IQ samples for TR to our data storage for labeling and future use during off-line training.

During phase 2 of the competition (2018), twenty technologies were participating in the competition and used the capabilities of the Colosseum for training their intelligent radios.

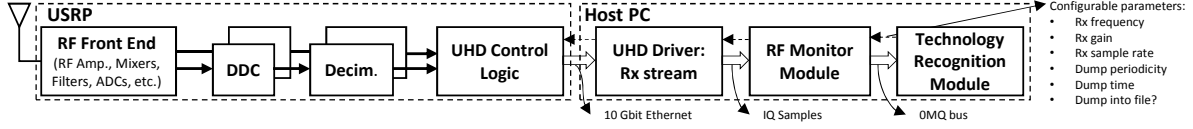


Fig. 5: RF Monitor module and its connections.

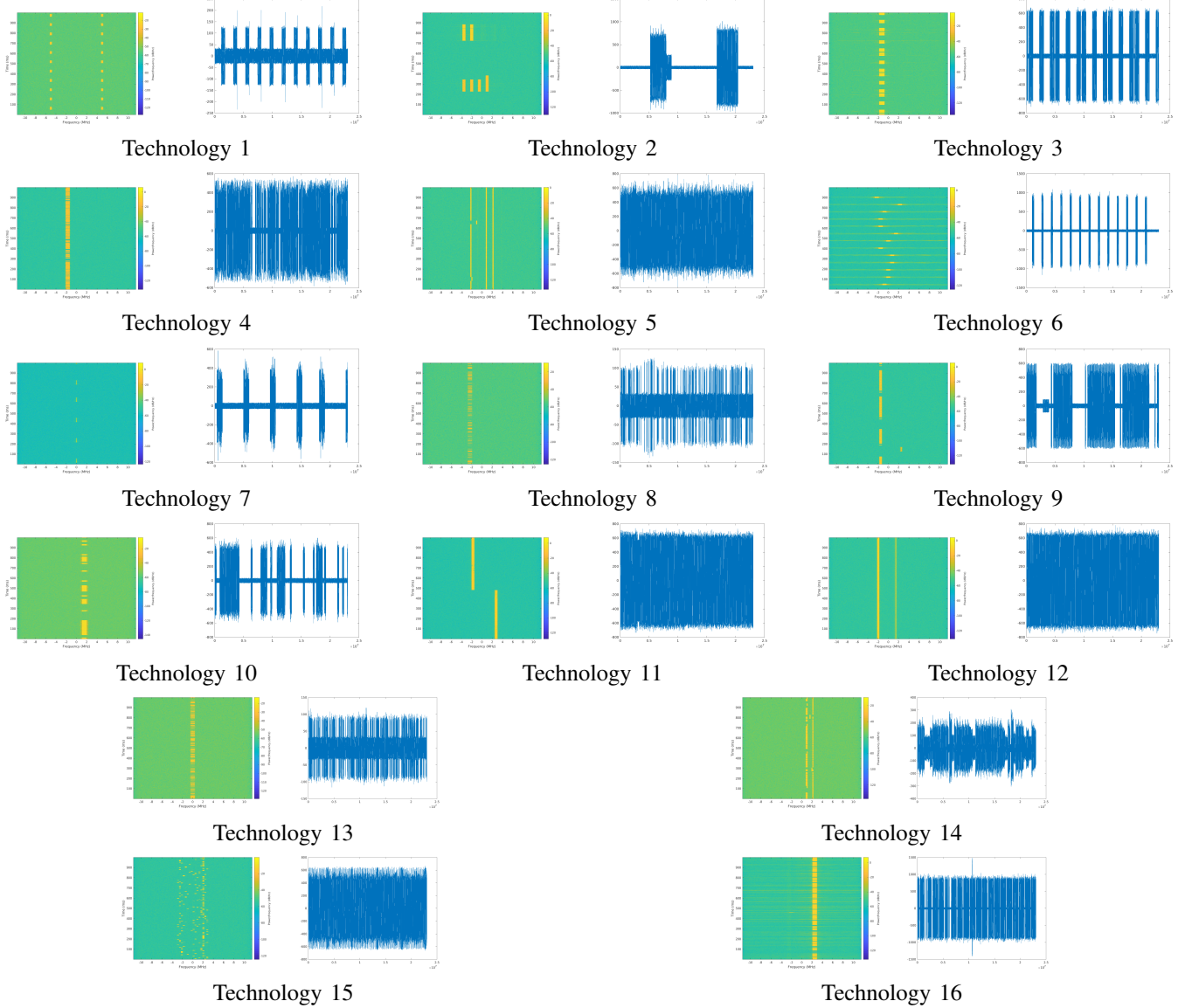


Fig. 6: Time and time-frequency signatures of the wireless technologies to be recognized

We played 55 games in a 6 Mhz bandwidth scenario with a constant 60-dB path loss among all the nodes. Each game was played against random technologies, and we managed to collect IQ samples of 16 out of 19 technologies (excluding us) and noise (idle period). The I and Q values are stored using 16-bit binary integers. The RF monitor block was set to collect samples at 23.04Mps, giving us 43.04 ns space between IQ samples. Figure 6 shows the resulting amplitude and spectrograms from some of the collected IQ samples for each unknown technology. A particular case was Technology 1 that was transmitting out-of-band.

Once the batches of IQ samples were collected, a couple of

those were visualized as spectrograms and labeled accordingly as different technologies. These limited examples are termed as the representative examples, which were used in the SL classification part of the proposed semi-supervised approach. After removing the noise of the captured signals using a Support Vector Machines (SVM) and subsequently normalize the whole data set in the range [0,1], the total size of the IQ samples data set was 93 GBs.

The whole data set was transformed into 11.3M examples, where each example corresponds to a pair of 1024 I and Q values. For the implementation of the proposed approach and the baseline, a data set with 1k labels per technology was

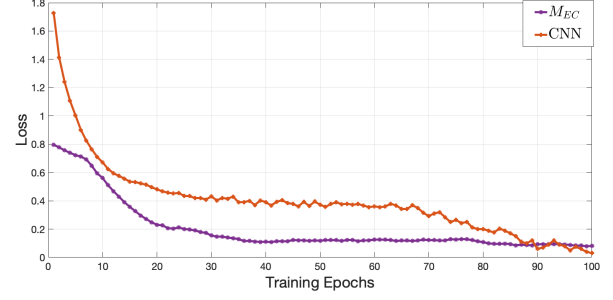
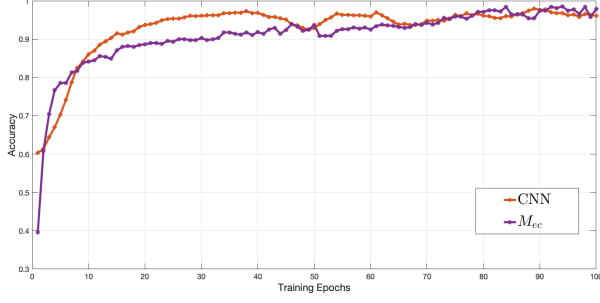


Fig. 7: Model training convergence: accuracy (left) and loss (right) curves of the validation data set

Type	Value
Total examples	11.3M
Size of example	1024 x 2 x 16 bits
Number of labels	17, 16 technologies + noise
Size data storage	93GB
Training and validation dataset	1k examples per label
Augmented dataset for SNR test	10k examples per label

TABLE I: Summary of the TR dataset.

used for training, validation, and test. Also, a second data set was created based on the 1k data set via data augmentation to emulate ten different SNR levels. The summary of the TR data set is given in Table I.

VI. RESULTS

In this section, the proposed SSL approach is compared against the CNN trained in a supervised way in terms of a) convergence performance, b) performance in the presence of different noise conditions, and c) labeling efficiency. For the comparison in the result section, we termed the proposed approach and the baseline as M_{EC} and CNN , respectively.

A. Algorithm Convergence

To evaluate the performance of DNN models, two performance metrics are often used: the accuracy, which measures the proportion of examples that the model can predict them correctly, and the loss, which quantify the inconsistency of the predicted value \hat{y} and the actual label y . The original data set of 1k samples per technology split into training (80%), validation (10%), and test (10%) data sets. In this evaluation, the whole training labeled data set was used for training both models. The impact of reducing the number of labels used for training is analyzed in Section VI-C. Figure 7 shows the validation accuracy and loss of the models with respect to the number of epochs. Here, we limit ourselves to showing only the validation results and not the training results because we want to show the generalizing performance. The proposed SSL model M_{EC} achieves similar convergence performance as the baseline model CNN trained using SL. It is important to notice that at epoch 50, when the weights of the encoder M_E are unfrozen for fine-tuning, a slight decrease in the accuracy for a couple of epochs is observed. This is expected because the model M_E needs to be modified for the new classification task, but later on, it keeps increasing the accuracy over time and reaching the same performance as the CNN model. The loss

curves decrease as expected, which elaborate correct learning over time. A smooth decaying loss function of M_{EC} shows that the selected hyperparameters, such as learning rate and regularization, help the optimizer to find an optimal point.

B. Model performance in the presence of noise

A good accuracy performance of ML models for the TR problem under different levels of noise validates the efficacy of the models in noisy environments. One way to achieve this is to apply data augmentation techniques to the available training data set, which can be achieved by processing the data set by including different levels of noise. However, models that can implicitly learn to be more tolerant of noise without requiring explicit training under noise are more powerful in real applications. To evaluate this performance, we generated a new data set by performing data augmentation on the original 1k data set with different levels of SNR. The concern of this augmented data set is to validate the performance of the trained models M_{EC} and CNN and show how they can cope in noisy environments. Figure 9 shows the accuracy of both models when we test the augmented data set on the trained model. The results show that the trained SSL model M_{EC} is more robust in noisy environments than the baseline CNN. While the CNN is trained with the objective of being accurate on predicting labels from examples that comes from the data distribution, the unsupervised training exploits the properties of denoising DAE. Although we do not use the augmented data set to train the models, which will increase their architecture size, we do use dropout layers to act as a regularizer. This selection has the same effect as adding noise to hidden layers [23]. The unsupervised learning step in the proposed SSL approach exploits the denoising property of DAE, which forces the trained model to be more resilient to noisy examples. Figure 8 shows the confusion matrices at -5 dB, 0 dB, and 5 dB of the two models. For SNR levels lower than -15dB, both networks are not able to learn any useful information from the raw IQ samples, and their accuracy is very low. Above this value, it is clear that the M_{EC} model is more robust than the CNN-based model. In fact, the accuracy at 0 dB is 3x better in our M_{EC} model.

C. Labeling efficiency

The previous evaluations assumed that the M_{EC} model had been trained using the same amount of labels as the CNN in order to have a fair comparison. Also, both the models are

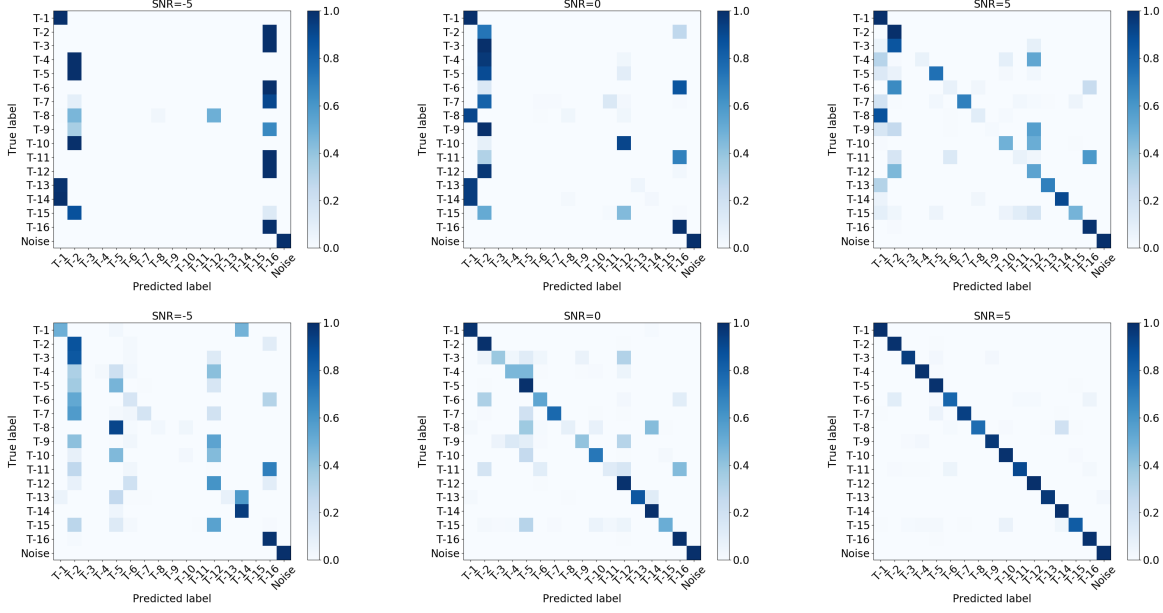


Fig. 8: Confusion matrices for CNN (top) and SSL algorithm using DAE (bottom) at different SNR

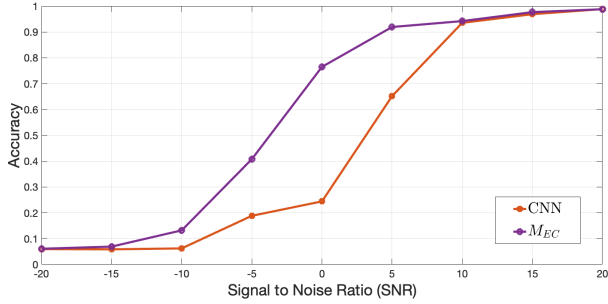


Fig. 9: Model accuracy at different SNR

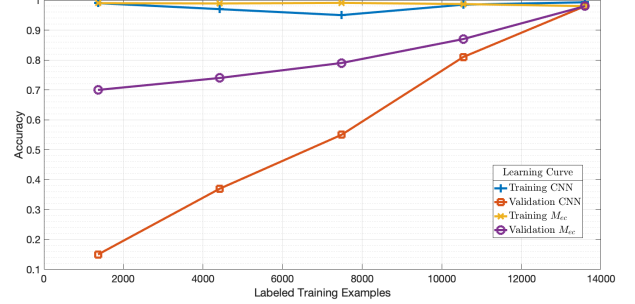


Fig. 10: Impact of the number of labels used during learning

fine-tuned by adjusting the hyperparameters individually for achieving high accuracy, i.e., up to 97% on the 1k data set. However, we aim to limit the required amount of labeled data set while still achieving good performance using our approach. The first aspect to verify on the DAE training is if the feature extraction task is performed correctly. This is fundamental in order to reduce the number of samples of the labeled data set to train the classifier. Figure 11 shows a random $q(t)$ example from the test data set and the signal reconstructed by the DAE. Although the signals are not entirely identical, it is clear that the encoder is learning essential features to perform such reconstruction correctly. Now, as the DAE training is unsupervised, the next step is to evaluate the impact of the number of labels during the supervised training. Figure 10 shows the impact of varying the number of labels available for training on the achieved training and validation accuracy of both the models. For this evaluation, we train the models with the labeled data sets of different sizes such as 10%, 33%, 55%, 77%, and 100% of the one used for the initial training. It is clear from the figure that the proposed approach, M_{EC} , takes advantage of the USL step to bootstrap the validation accuracy with a limited number of labels. More importantly,

the proposed approach achieves an accuracy of $\geq 70\%$ with only 10% of the total number of labels, which is translated to 4.6 times better accuracy than the CNN model using the same amount of labeled data. In other words, M_{EC} generalizes better to unseen data than the CNN with a reduced set of labels. Once there is more labeled data available for training, the CNN model increases its validation accuracy. On the other hand, note that the training accuracy of both the models achieves 100%. This behavior, i.e., overfitting the training data set, it is expected since the DL models were designed and optimized for using the whole data set. However, while M_{EC} was able to have an increasing validation accuracy with the reduced data set, the CNN model memorized the reduced data set, and it did not extract features to generalize unseen (validation) data.

VII. CONCLUSIONS

Technology Recognition (TR) will play an essential role in how new wireless technologies make decisions in order to use the available spectrum efficiently and coexist with any new, legacy, and even unknown technologies. In this paper, we have proposed a novel Semi-supervised Learning (SSL)

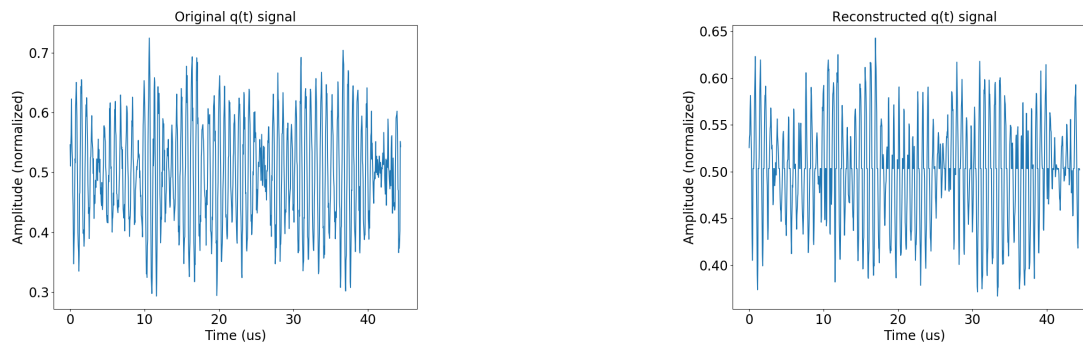


Fig. 11: DAE reconstruction: Original $q(t)$ signal (left) and reconstructed $q(t)$ signal (right)

approach for TR that minimizes the need for labeling large data sets of spectrum data. In addition, the proposed approach requires only raw IQ samples, which can easily be acquired from low-cost sensing devices.

The evaluation illustrates that the proposed approach can achieve an accuracy of $\geq 70\%$ with only 10% of the total number of labels, which is translated to 4.6 times better accuracy than the considered baseline CNN model using the same amount of labeled data. Besides, we found that the resulting DL model is more robust under corrupted input, e.g., noisy signals, than the CNN-based model, with up to 2x better accuracy at SNR levels of -5dB and up to 3x at 0dB. Future work will mainly cover the integration of the TR module into our stack to enhance our ML decision engine with useful information for smart spectrum access.

REFERENCES

- [1] Cisco, "Cisco visual networking index: global mobile data traffic forecast update, 2017-2022," <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf>.
- [2] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2013.
- [3] E. Karami, O. A. Dobre, and N. Adnani, "Identification of gsm and lte signals using their second-order cyclostationarity," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, May 2015, pp. 1108–1112.
- [4] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.
- [5] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, March 2009.
- [6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [7] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. The MIT Press, 2010.
- [8] DARPA, "World's most powerful emulator of radio-signal traffic opens for business," <https://www.darpa.mil/news-events/2017-04-21>, accessed: 2019-03-01.
- [9] Z. Zhu and A. K. Nandi, *Modulation Classification Features*. John Wiley and Sons, Ltd, 2014.
- [10] O. A. Dobre, "Signal identification for emerging intelligent radios: classical problems and new challenges," *IEEE Instrumentation Measurement Magazine*, vol. 18, no. 2, pp. 11–18, 2015.
- [11] M. Shi, A. Laufer, and Y. B. and, "Fourth order cumulants in distinguishing single carrier from ofdm signals," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, Nov 2008, pp. 1–6.
- [12] A. Al-Habashna, O. A. Dobre, R. Venkatesan, and D. C. Popescu, "Second-order cyclostationarity of mobile wimax and lte ofdm signals and application to spectrum awareness in cognitive radio systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 1, pp. 26–42, 2012.
- [13] E. Karami and O. A. Dobre, "Identification of sm-ofdm and al-ofdm signals based on their second-order cyclostationarity," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 942–953, 2015.
- [14] M. Firdaoussi, H. Ghennioui, and M. El Kamili, "Recognition of ofdm and scfd signals based on the generalized mean ambiguity function," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016, pp. 230–234.
- [15] A. Bouzegzi, P. Ciblat, and P. Jallon, "Maximum likelihood based methods for ofdm intercarrier spacing characterization," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008, pp. 1–5.
- [16] M. Kulin, T. Kazaz, I. Moerman, and E. De Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18 484–18 501, 2018.
- [17] N. Bitar, S. Muhammad, and H. H. Refai, "Wireless technology identification using deep convolutional neural networks," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6.
- [18] S. Yi, H. Wang, W. Xue, X. Fan, L. Wang, J. Tian, and R. Matsukura, "Interference source identification for ieee 802.15.4 wireless sensor networks using deep learning," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1–7.
- [19] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06. Cambridge, MA, USA: MIT Press, 2006, pp. 153–160. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2976456.2976476>
- [20] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3563–3593, Jan. 2014. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2627435.2750359>
- [21] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. MIT Press, 1995.
- [22] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks*, C. Jayne and L. Iliadis, Eds. Cham: Springer International Publishing, 2016, pp. 213–226.
- [23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
- [25] F. Chollet et al. Keras. [Online]. Available: <https://keras.io>
- [26] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [27] DARPA. (2019) System specification document. [Online]. Available: <https://www.spectrumcollaborationchallenge.com/wp-content/uploads/2018/12/System-Specification-Document.pdf>